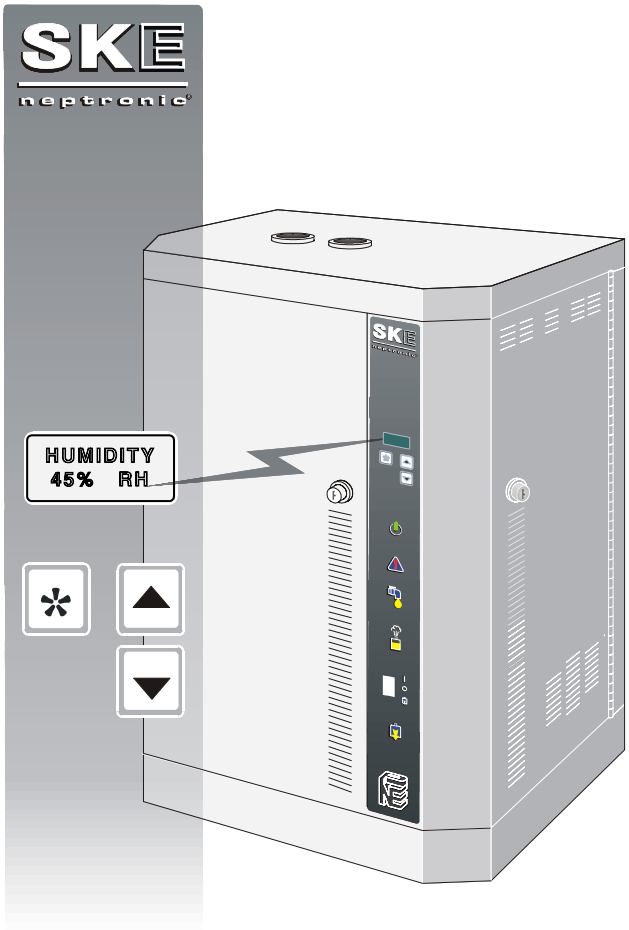
neptronic®

# SKE Series
# Steam Humidifier

## BACnet® Communication Module
## User Guide




CE

BACnet™

SKE-BACNET – Rev.: 081104

# Table of content



**Other related documents:**

1. Installation instructions
2. Wiring diagram

# Introduction

This document provides a Users Guide for using the NEP PIC Communications Module (NEPIC). This product provides a BACnet® network interface between BACnet® client devices and NEP Humidifier Series devices, specifically for SKE model. NEPIC uses the BACnet® Master Slave/Token Passing (MS/TP) protocol at the BACnet® MAC layer.

This document assumes you're familiar with BACnet® and BACnet® terminology.

# BACnet® Requirements

### Performance

The NEPIC uses a synchronous implementation for BACnet® messages. Each BACnet® confirmed service request is answered as quickly as possible without using Reply Postponed. In particular, MS/TP implementation performs within Tusage_delay of 15ms in order to assure Tusage_timeout values within 20ms.

### Support for MS/TP

The NEPIC supports a Full Master Node state machine for MS/TP. Max_Master and the MS/TP MAC address shall be configurable through WriteProperty service to the device object. A default MAC address of 254 shall be recognized when a configuration dip switch is set to configure mode. Two other dip switches shall determine MS/TP baud rate 9600, 19200, 38400, and 76800. When in the configure mode WriteProperty service requests may be directed to MAC address 254 using the wildcard Device instance (4194303 decimal/0x3FFFFF hex) as a means of configuring all other parameters for the device. Also, when in the configuration mode the MS/TP MAC address and the Device Instance shall be configurable through the Unit's Keypad.

### BIBB Support

The NEPIC generally behaves as a B-ASC type profile server. The following specific BIBBs are supported per their relevant definitions in Annex K to BACnet®:

DS-RP-B, DS-WP-B, DM-DDB-B, DM-DOB-B, DM-DCC-B

### Object Support (in general)

The NEPIC supports a table-based fixed list of BACnet®-visible values which appear as Present_Values of various BACnet® standard object types, in addition to a Device object.

### Alarms

Although the NEPIC supports the ability to indicate various alarm conditions through value changes in properties of several of its objects, it **does not** generate BACnet® Event Notifications.

# Device Object

The following table lists all the BACnet® properties supported for the device object. W? indicates if the property is writable using the BACnet® WriteProperty service.

## Device

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *programmable where the instance part of the Object_Identifier is in the range of 0-4194302*. The device instance must be unique system-wide. The default value for the device instance=153000 (Vendor_Identifier*1000) | W |
| Object_Name | *programmable up to 32 characters.* The device name must be unique system-wide. The default value= "NEP Humidifier 153000" where 153000 is the Vendor_Identifier*1000 | W |
| Object_Type | 8 | |
| System_Status | always *operational* | |
| Vendor_Identifier | always 153 | |
| Vendor_Name | always “National Environmental Products Ltd | |
| Model_Name | “SKE” for example | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Firmware_Revision | currently “1.00” | |
| Application_Software_Version | currently “1.00” | |
| Protocol_Version | always 1 | |
| Protocol_Revision | always 2 | |
| Max_APDU_Length_Accepted | always 107 | |
| Segmentation_Supported | always *none* | |
| APDU_Timeout | always 0 | |
| Number_of_APDU_Retries | always 0 | |
| Protocol_Services_Supported | aways 0x00 0x09 0x40 0x00 0xE0 (i.e. a *bitstring* in BACnet® order | |
| Protocol_Object_Types_Supported | aways 0xB4 0x84 0x10 0x00 (i.e. a *bitstring* in BACnet® order | |
| Object_List | *per the standard*. Because of restrictions on the size of the transmit buffers, the entire Object_List cannot be returned at once, rather the Object_List must be read, one-at-a-time | |
| Local_Time | *per the standard*, if the unit supports a RTC | |
| Local_Date | *per the standard*, if the unit supports a RTC | |
| Device_Address_Binding | always *empty* | |
| Max_Master | *programmable in the range of 0-127.* Default value=127 | W |
| Max_Info_Frames | always 1 | |
| Proprietary property #1000 | *programmable.* This proprietary property represents the MS/TP MAC address in the range of (0-254). Values 128 to 254 represent MS/TP non-token-passing slave devices. Default value=0 | W |

# Objects

A complete list of all BACnet® objects for the NEPIC are listed in the following section. There are a total of 84 BACnet® objects per NEPIC consisting of the following types:

- 1 Device
- 7 Analog Inputs (AI)
- 25 Analog Values (AV)
- 15 Binary Inputs (BI)
- 30 Binary Values (BV)
- 6 Multistate Values (MSV)

The Device Object has already been described. The following tables list all the BACnet® properties supported for each object type. Most of the properties are locked in. The exceptions are Present_Values, which represent the dynamic operating values of the device, and the Status_Flags, Event_States and Reliabilitys which reflect the availability of the Present_Values. Unless otherwise specified, properties are not changeable.

## Analog Inputs

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *see Object Table* | |
| Object_Name | *see Object Table* | |
| Object_Type | 0 | |
| Present_Value | *see Object Table* | |
| Max_Present_Value | *see Object Table* | |
| Min_Present_Value | *see Object Table* | |
| Description | *see Object Table* | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Event_State | If the *fault* bit of the Status_Flags is set, this property's value is *fault*, otherwise it's *normal* | |
| Reliability | If the *fault* bit of the Status_Flags is set, this property's value is *unreliable_other*, otherwise it's *no_fault_detected* | |
| Out_of_Service | Always *false* | |
| Units | *see Object Table* | |

## Analog Values

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *see Object Table* | |
| Object_Name | *see Object Table* | |
| Object_Type | 2 | |
| Present_Value | *see Object Table* | W |
| Max_Present_Value | *see Object Table* | |
| Min_Present_Value | *see Object Table* | |
| Description | *see Object Table* | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Event_State | If the *fault* bit of the Status_Flags is set, this property's value is *fault*, otherwise it's *normal* | |
| Reliability | If the *fault* bit of the Status_Flags is set, this property's value is *unreliable_other*, otherwise it's *no_fault_detected* | |
| Out_of_Service | Always *false* | |
| Units | *see Object Table* | |

## Binary Inputs

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *see Object Table* | |
| Object_Name | *see Object Table* | |
| Object_Type | 3 | |
| Present_Value | *see Object Table* | |
| Description | *see Object Table* | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Event_State | If the *fault* bit of the Status_Flags is set, this property's value is *fault*, otherwise it's *normal* | |
| Reliability | If the *fault* bit of the Status_Flags is set, this property's value is *unreliable_other*, otherwise it's *no_fault_detected* | |
| Out_of_Service | Always *normal* | |
| Polarity | *see Object Table* | |
| Active_Text | *see Object Table* | |
| Inactive_Text | *see Object Table* | |

## Binary Values

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *see Object Table* | |
| Object_Name | *see Object Table* | |
| Object_Type | 5 | |
| Present_Value | *see Object Table* | W |
| Description | *see Object Table* | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Event_State | If the *fault* bit of the Status_Flags is set, this property's value is *fault*, otherwise it's *normal* | |
| Reliability | If the *fault* bit of the Status_Flags is set, this property's value is *unreliable_other*, otherwise it's *no_fault_detected* | |
| Out_of_Service | Always *normal* | |
| Polarity | *see Object Table* | |
| Active_Text | *see Object Table* | |
| Inactive_Text | *see Object Table* | |

## Multistate Values

| *property* | *value* | *W?* |
|---|---|---|
| Object_Identifier | *see Object Table* | |
| Object_Name | *see Object Table* | |
| Object_Type | 19 | |
| Present_Value | *see Object Table* | W |
| Description | *see Object Table* | |
| Status_Flags | If the *fault* bit is set, it indicates that the Present_Value is unavailable and is unreliable | |
| Event_State | If the *fault* bit of the Status_Flags is set, this property's value is *fault*, otherwise it's *normal* | |
| Reliability | If the *fault* bit of the Status_Flags is set, this property's value is *unreliable_other*, otherwise it's *no_fault_detected* | |
| Out_of_Service | Always *normal* | |
| Number_of_States | *see Object Table* | |

# SKE Humidifier Object Table

The SKE humidifier series of controllers use the following BACnet® object table. The *type* is the BACnet® Object type, the *instance* is the BACnet® Object. *W?* indicates whether the Present_Value property is writable. Together the *type* and *instance* form the BACnet® Object_Identifier for an object according to the following C-language algorithm:

- object_identifier=(unsigned long)((unsigned long)*type*<<22)+*instance*

## Analog Input

| *type* | *inst* | *Object_Name* | *Description* | *range of Present_Value* | *W?* |
|---|---|---|---|---|---|
| AI | 1 | Humidity_Out | Humidity Output | 0-100 %RH | |
| AI | 2 | Water_Level | Water level | 0-120% | |
| AI | 3 | Delay_to_Drain | Delay until Drain | 0-100Hrs | |
| AI | 4 | Water_Temp | Water Temperature | 0-260 °C or 32-500 °F | |
| AI | 5 | Chimney_Temp | Chimney Temperature | 0-260 °C or 32-500 °F | |
| AI | 6 | SSR_Temp | Solid State Relay Temperature | 0-260 °C or 32-500 °F | |
| AI | 7 | Op_Delay | Operation Delay | 0-5000Hrs | |

## Analog Values

| *type* | *inst* | *Object_Name* | *Description* | *range of Present_Value* | *W?* |
|---|---|---|---|---|---|
| AV | 1 | Humidity_Demand | Humidity Demand | 0-100 %RH | W |
| AV | 2 | Room_Humidity | Room Humidity | 0-100 %RH | W |
| AV | 3 | Duct_Humidity | Duct Humidity | 0-100 %RH | W |
| AV | 4 | Drain_Delay | Drain Delay | 1-100Hrs | W |
| AV | 5 | Svc_Delay | Service Delay | 1-5000Hrs | W |
| AV | 6 | EOS_Delay | End of season delay | 1-250Hrs | W |
| AV | 7 | PID_Loop_KP | PID Loop gain KP | 0-255 | W |
| AV | 8 | PID_Loop_KI | PID Loop gain KI | 0-255 | W |
| AV | 9 | PID_Loop_KD | PID Loop gain KD | 0-255 | W |
| AV | 10 | Ctl_Band | Control band | 0-20.0% | W |
| AV | 11 | Ctl_SP | Control set point | 0-100%RH | W |
| AV | 12 | Extern_SP_Min | External set point minimum | 0-100%RH | W |
| AV | 13 | Extern_SP_Max | External set point maximum | 0-100%RH | W |
| AV | 14 | Duct_Hum_HL_SP | Duct hum. high limit set point | 50-100%RH | W |
| AV | 15 | Fan_Min_On_Time | Minimum ON time for fan | 2-20 minutes | W |
| AV | 16 | Min_Water_Temp | Min water temp. in On mode | 10-90 °C or 50-194 °F (0 -> Mode is Off) | W |
| AV | 17 | Antifreeze_Temp | Anti-freeze mode temp | 4-10 °C or 39-50°F (0 -> Mode is Off) | W |
| AV | 18 | Room_Hum_Offset | Room Humidity offset | ±0-20 %RH | W |
| AV | 19 | Duct_Hum_Offset | Duct Humidity offset | ±0-20 %RH | W |
| AV | 20 | Water_Temp_Offset | Water Temperature offset | ±0-20 °C or ±0-40 °F | W |
| AV | 21 | Water_Lvl_Offset | Water Level offset | ±0-20% | W |
| AV | 22 | SSR_Temp_Offset | SSR Temperature offset | ±0-20 °C or ±0-40 °F | W |
| AV | 23 | Lock_On_Cap | Lock On capacity | 10-100% | W |
| AV | 24 | Max_Out_Limit | Maximum output limit | 10-100% | W |
| AV | 25 | Outside_Temp | Outside Temperature | -40 - 40 °C or -40 - 104 °F | W |

# SKE Humidifier Object Table

## Binary input

| *type* | *inst* | *Object_Name* | *Description* | *range of Present_Value* | *W?* |
|---|---|---|---|---|---|
| -- | -- | -- | Actual Operating Status: | | |
| BI | 1 | Power_Status | Power Status | 0=OFF, 1=ON | |
| BI | 2 | Demand | Demand | 0=NO, 1=YES | |
| BI | 3 | Auto_Drain | Auto Drain | 0=NO, 1=YES | |
| BI | 4 | Manual_Drain | Manual Drain | 0=NO, 1=YES | |
| BI | 5 | Alrm_Not_Critical | Alarm not critical (unit running) | 0=NO, 1=YES | |
| BI | 6 | Alarm_Critical | Alarm critical (unit not running) | 0=NO, 1=YES | |
| BI | 7 | EOS_Delay_Status | End of Season Delay | 0=NO, 1=YES | |
| -- | - | | Misc. input Status: | | |
| BI | 8 | Fan_Proof | Fan proof Status | 0=OFF, 1=ON | |
| BI | 9 | Hi_Lim_Status | Hi-Limit Status | 0=OFF, 1=ON | |
| BI | 10 | Interlock | Interlock | 0=OFF, 1=ON | |
| BI | 11 | Power_Fuse | Power Fuse (24VAC) | 0=OFF, 1=ON | |
| BI | 12 | Klixxon | Klixxon | 0=OFF, 1=ON | |
| BI | 13 | Ext_Therm_Fuse | Thermal fuse (external 24VAC) | 0=OFF, 1=ON | |
| BI | 14 | Int_Therm_Fuse | Thermal fuse (internal 24VDC) | 0=OFF, 1=ON | |
| BI | 15 | Foam_Probe | Foam Probe | 0=OFF, 1=ON | |

# SKE Humidifier Object Table

## Binary Values

| *type* | *inst* | *Object_Name* | *Description* | *range of Present_Value* | *W?* |
|---|---|---|---|---|---|
| -- | -- | -- | Alarm Indications: | | |
| BV | 1 | Over_Temp_ON | Over temperature contact ON | 0=Normal, 1=Alarm | W |
| BV | 2 | Pwr_Fuse_Open | Power fuse (24VAC) open | 0=Normal, 1=Alarm | W |
| BV | 3 | Ext_Th_Fuse_Open | Thermal Fuse open (24VAC) | 0=Normal, 1=Alarm | W |
| BV | 4 | Int_Th_Fuse_Open | Thermal Fuse open (24VDC) | 0=Normal, 1=Alarm | W |
| BV | 5 | Hi_Duct_Hum | High humidity level in the duct | 0=Normal, 1=Alarm | W |
| BV | 6 | Low_Input_Volt | Input voltage too low | 0=Normal, 1=Alarm | W |
| BV | 7 | Bad_Tank_Sensor | Tank sensor defective | 0=Normal, 1=Alarm | W |
| BV | 8 | Bad_SSR_Sensor | SSR sensor defective | 0=Normal, 1=Alarm | W |
| BV | 9 | Bad_Water_Sensor | Water level probe defect | 0=Normal, 1=Alarm | W |
| BV | 10 | Foam_Sensor_ON | Foam Sensor ON | 0=Normal, 1=Alarm | W |
| BV | 11 | Low_Tank_Temp | Tank temperature too low (freeze) | 0=Normal, 1=Alarm | W |
| BV | 12 | Hi_Tank_Temp | Tank temperature too high | 0=Normal, 1=Alarm | W |
| BV | 13 | Hi_SSR_Temp | SSR temperature too high | 0=Normal, 1=Alarm | W |
| BV | 14 | Bad_Fill_Tank | Defective filling tank | 0=Normal, 1=Alarm | W |
| BV | 15 | Bad_Refill_Tank | Defective refill tank (too long) | 0=Normal, 1=Alarm | W |
| BV | 16 | Bad_Drain_Tank | Defective draining tank | 0=Normal, 1=Alarm | W |
| BV | 17 | Duct_Hi_Lim_CO | Duct Hi limit cut-out | 0=Normal, 1=Alarm | W |
| BV | 18 | Interlock_In_Open | Interlock Input Open | 0=Normal, 1=Alarm | W |
| BV | 19 | Hum_Clean_Time | Humidifier started cleaning period | 0=Normal, 1=Alarm | W |
| BV | 20 | Hum_Svc_Time | Humidifier exceeded service time | 0=Normal, 1=Alarm | W |
| BV | 21 | Demand_SP_Type | Demand-Set point type | 0=Voltage, 1=Current | W |
| BV | 22 | Demand_SP_Span | Demand-Set point span | 0=2-10VDC, 1=0-10VDC | W |
| BV | 23 | Humidity_In_Type | Humidity input type | 0=Voltage, 1=Current | W |
| BV | 24 | Humidity_In_Span | Humidity input span | 0=2-10VDC, 1=0-10VDC | W |
| BV | 25 | Hi_Lim_In_Type | Hi-limit humidity input type | 0=Voltage, 1=Current | W |
| BV | 26 | Hi_Lim_In_Span | Hi-limit humidity input span | 0=2-10VDC, 1=0-10VDC | W |
| -- | -- | -- | System Option | | |
| BV | 27 | Temp_Units | Temperature Units | 0=Celsius, 1=Fahrenheit | W |
| BV | 28 | Outside_Temp_Comp | Outside Temperature Compensation | 0=OFF, 1=ON | W |
| BV | 29 | Alarm_Beep | Alarm Beep | 0=OFF, 1=ON | W |
| BV | 30 | Reload_Factory | Reload Factory default value | 0=NO, 1=YES | W |

### Multistate Values

| *type* | *inst* | *Object_Name* | *Description* | *range of Present_Value* | *W?* |
|---|---|---|---|---|---|
| MSV | 1 | Ctl_Mode | Control Mode | 1=Local Demand from Analog Input 1<br>2=Internal PID calculation<br>3=Demand from remote communication port (e.g. BACnet®) | W |
| MSV | 2 | Operation_Mode | Operation Mode | 1=Set unit Off<br>2=Set unit On<br>3=Set unit in drain mode | W |
| MSV | 3 | Unit_Display_Lang | Unit Display Language | 1=English (Default)<br>2=French<br>3=Spanish | W |
| MSV | 4 | SP_Source | Set point Source | 1=Local analog input 1<br>2=Local Internal digital value<br>3=Remote from communication port (e.g. BACnet®) | W |
| MSV | 5 | Room_Hum_Source | Room Humidity source | 1=Local analog input 2<br>2=Remote from communication port (e.g. BACnet®) | W |
| MSV | 6 | Duct_Hum_Source | Duct Humidity source | 1=Not used<br>2=Local analog input 3<br>3=Remote from communication port (e.g. BACnet®) | W |

## Mode

Normally the NEPIC is in the operational mode. The NEPIC can be placed in the configuration mode at any time by throwing a single dip switch.

| *mode* | *Switch 1* |
|---|---|
| operational | OFF |
| configuration | ON |

The difference between configuration mode and operational mode is explained in detail later in this section. The NEPIC can be put into and out of the configuration mode at any time. When the unit detects a change in the position of switch 1, it automatically restarts itself in the appropriate mode.

## Baud Rate

The baud rate for the BACnet® MS/TP is configurable through a pair of dip switches. The following table identifies the baud rates used according to the switch settings:

| *baud rate* | *Switch 2* | *Switch 3* |
|---|---|---|
| 9600 | OFF | OFF |
| 19200 | ON | OFF |
| 38400 | OFF | ON |
| 76800 | ON | ON |

Please note, that you ***must*** restart the unit in order for a change of baud rate to take effect.

## Configurable BACnet® Properties

The following four BACnet® properties are configurable and in fact most likely will need to be changed to guarantee uniqueness of each device in a BACnet® system:

- Device Object.Object_Identifier *
- Device Object.Object_Name
- Device Object.Max_Master
- Device Object.*proprietary property #1000* (which will be called *MSTP_MACaddress* for the remainder of this section)

* Note: Because the Device's Object_Identifier is a combination of the Device Object_Type (8) and the Device_Instance (0-4194302) it's decimal or hexadecimal representation tends to be incomprehensible. Even the simple/easy-to-understand Device_Instance=1000 has an equivalent Object_Identifier of 0x020003E8 hexadecimal or 33555432 decimal. So, while it's the device's Object_Identifier property that can be changed using a BACnet® WriteProperty service, this document will talk mostly about Device_Instances.

## Getting Started

The four configurable BACnet® Device Object properties have two sets of "default" settings, the factory setting and the configuration mode setting and are identified in the following table:

| *property* | *factory value* | *configuration mode value* |
|---|---|---|
| Device_Instance | 153000* | 153000* |
| Object_Name | "NEP Humidifier 4000" | "NEP Humidifier 4000" |
| Max_Master | 127 | 127 |
| *MSTP_MACaddress* | 0 | 254 |

* Note: These values are NEP's BACnet® Vendor_Identifier*1000.

Prior to the first time the NEPIC is powered on, you'll have to know two things:

1. What's the baud rate of the MS/TP network?
2. Is there already an MS/TP unit on that network with the MAC address=0 and the Device Instance=153000?

Once the answer to Question 1 is known, the dip switches 2 and 3 must be set up accordingly.

If the answer to Question 2 is *no* (there is no MS/TP MAC address 0), then you can start up the NEPIC with the dip switch 1 in the OFF (operational) position. In this mode, the factory settings are in effect and the NEPIC will be MS/TP token-passing master with a MAC address=0 and Device Instance=153000.

If the answer to Question 2 is *yes* (there already is an MS/TP MAC address 0 and/or there already is a Device Instance 153000), then you will need to start up the NEPIC with the dip switch 1 in the ON (configuration) position. In this mode, the configuration mode settings are in effect and the NEPIC will be MS/TP non-token-passing slave with a MAC address 254.

## Configuring in the Operational Mode

The NEPIC can be configured from a BACnet® client device using the BACnet® WriteProperty service at any time, while in the operational mode. In other words, the Device_Instance, Object_Name, Max_Master and *MSTP_MACaddress* can be changed "hot" with the changes taking effect immediately and without having to restart the NEPIC.

## Configuring in the Configuration Mode

If the NEPIC is put into the configuration mode while it is "hot", the NEPIC will be automatically reset with the Device_Instance, Object_Name, Max_Master and *MSTP_MACaddress* all containing the *configuration mode values*. In this mode, the NEPIC will **not** act as an MS/TP token-passer and will be silent until it is addressed by a BACnet® client. As a side effect, it will also not be able to respond to BACnet® WhoIs services with Iam services, so BACnet® clients will not be able to find out its Device_Instance automatically. In the configuration mode, any of the above properties can be changed by using either the default Device_Instance or the wild card Device_Instance (4194303 decimal or 0x3FFFFF hex). Use of the wild card Device_Instance obviates the need to know the NEPIC's real Device_Instance in case it conflicts with another Device in the system. While in the configuration mode, only the Device Object is available to BACnet® clients through the ReadProperty and Write Property services. All other objects (i.e. AI's, etc) are not available.

As an alternative to using the BACnet® WriteProperty service to change the Device_Instance and/or the *MSTP_MACaddress*, the SKE's keypad can be used to change those two properties, but only while in the configuration mode.

Changes to the Device Object, whether made by using the WriteProperty of the keypad, do not take effect until the NEPIC is restarted in the operational mode.

## Selecting the MS/TP MAC address and Max_Master

Some care must be taken in setting the MS/TP MAC address and Max_Master property.

First, the MAC address must be unique on the entire MS/TP network. However, having a unique MAC address and a high baud rate does not guarantee efficient operation of the NEPIC (or other MS/TP units for that matter) on the MS/TP network. Some MAC address and Max_Master combinations are more efficient than others. BACnet® requires token-passing units to occasionally "poll" for other masters based on the value of MAC address and Max_Master. So, a "poor" combination of MAC addresses and Max_Masters can lead to slow networks in which there's a lot of wasted time polling for masters that are not present and never will be. In fact, unless there are 126 other units on the MS/TP network, the default Max-Master=127 is most likely a poor choice for the NEPIC. Having said that, Max-Master=127 has been chosen as the default to insure that any master, specifically a BACnet® client, can be found when the NEPIC is first started.

So, considering the following simple two-unit examples:

*Example 1:*

- MAC=0. Max_Master=127
- MAC=1, Max_Master=127

This example is slow and inefficient because both Max_Master=127. Everytime either unit is required to find another master units it has to poll 126 units until it finds the right one to pass the token to.

*Example 2:*

- MAC=0. Max_Master=5
- MAC=5, Max_Master=5

This example is better but is still slower than it could be. The Max_Master is set to the most efficient value, however because of the gap between the two MAC addresses, each unit must poll 4 units until it finds the right one to pass the token to.

*Example 3:*

- MAC=0. Max_Master=1
- MAC=2, Max_Master=2

This example is actually an incorrect configuration, in that MAC=0 will never find MAC=2 because it will never poll for the master MAC address=2.

*Example 4:*

- MAC=0. Max_Master=1
- MAC=1, Max_Master=1

This example is the most efficient, since each unit must poll only 1 other unit until it finds the right one to pass the token to.

As a general guideline, the most efficient set up for an MS/TP network is one in which the units are consecutively numbered starting at MAC address 0 and all have Max_Master=the maximum MAC address in the system. If consecutive numbering is not possible, then the next most efficient set up is one in which all units have have Max_Master=the maximum MAC address in the system.

# Personal notes

# Personal notes

**National Environmental Products Ltd.**

400 Bd Lebeau, Montreal, Qc, H4N 1R6, CANADA

***www.neptronic.com***

Toll free in *North America*: 1 800 361-2308
Tel.: (1) 514-333-1433
Fax: (1) 514-333-3163
Customer service Fax: 514-333-1091
Business hours: from Monday to Friday, 8:00am to 5:00pm (Eastern time)